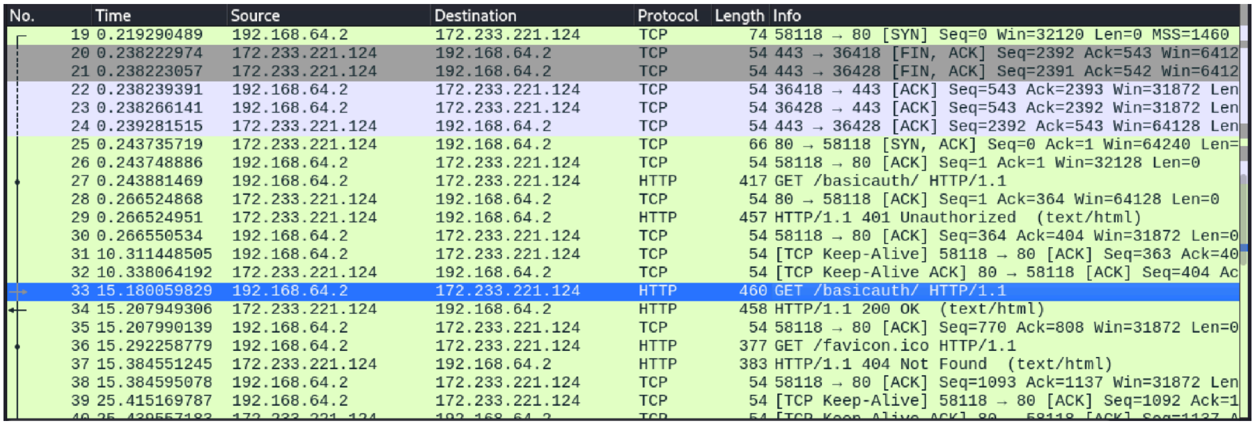
Lucky Beulla Muhoza
CS338: Computer Security
Prof. Jeff Ondich

September 25th, 2024

HTTP's Basic Authentication: A Story



No.	Time	Source	Destination	Protocol	Length	Info
19	0.219290489	192.168.64.2	172.233.221.124	TCP	74	58118 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460
20	0.238222974	172.233.221.124	192.168.64.2	TCP	54	443 → 36418 [FIN, ACK] Seq=2392 Ack=543 Win=6412
21	0.238223057	172.233.221.124	192.168.64.2	TCP	54	443 → 36428 [FIN, ACK] Seq=2391 Ack=542 Win=6412
22	0.238239391	192.168.64.2	172.233.221.124	TCP	54	36418 → 443 [ACK] Seq=543 Ack=2393 Win=31872 Len
23	0.238266141	192.168.64.2	172.233.221.124	TCP	54	36428 → 443 [ACK] Seq=543 Ack=2392 Win=31872 Len
24	0.239281515	172.233.221.124	192.168.64.2	TCP	54	443 → 36428 [ACK] Seq=2392 Ack=543 Win=64128 Len
25	0.243735719	172.233.221.124	192.168.64.2	TCP	66	80 → 58118 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=
26	0.243748886	192.168.64.2	172.233.221.124	TCP	54	58118 → 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0
27	0.243881469	192.168.64.2	172.233.221.124	HTTP	417	GET /basicauth/ HTTP/1.1
28	0.266524868	172.233.221.124	192.168.64.2	TCP	54	80 → 58118 [ACK] Seq=1 Ack=364 Win=64128 Len=0
29	0.266524951	172.233.221.124	192.168.64.2	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
30	0.266550534	192.168.64.2	172.233.221.124	TCP	54	58118 → 80 [ACK] Seq=364 Ack=404 Win=31872 Len=0
31	10.311448505	192.168.64.2	172.233.221.124	TCP	54	[TCP Keep-Alive] 58118 → 80 [ACK] Seq=363 Ack=40
32	10.338064192	172.233.221.124	192.168.64.2	TCP	54	[TCP Keep-Alive ACK] 80 → 58118 [ACK] Seq=404 Ac
33	15.180059829	192.168.64.2	172.233.221.124	HTTP	460	GET /basicauth/ HTTP/1.1
34	15.207949306	172.233.221.124	192.168.64.2	HTTP	458	HTTP/1.1 200 OK (text/html)
35	15.207990139	192.168.64.2	172.233.221.124	TCP	54	58118 → 80 [ACK] Seq=770 Ack=808 Win=31872 Len=0
36	15.292258779	192.168.64.2	172.233.221.124	HTTP	377	GET /favicon.ico HTTP/1.1
37	15.384551245	172.233.221.124	192.168.64.2	HTTP	383	HTTP/1.1 404 Not Found (text/html)
38	15.384595078	192.168.64.2	172.233.221.124	TCP	54	58118 → 80 [ACK] Seq=1093 Ack=1137 Win=31872 Len
39	25.415169787	192.168.64.2	172.233.221.124	TCP	54	[TCP Keep-Alive] 58118 → 80 [ACK] Seq=1092 Ack=1
40	25.439557183	172.233.221.124	192.168.64.2	TCP	54	[TCP Keep-Alive ACK] 80 → 58118 [ACK] Seq=1137 A

Wireshark capture screenshot

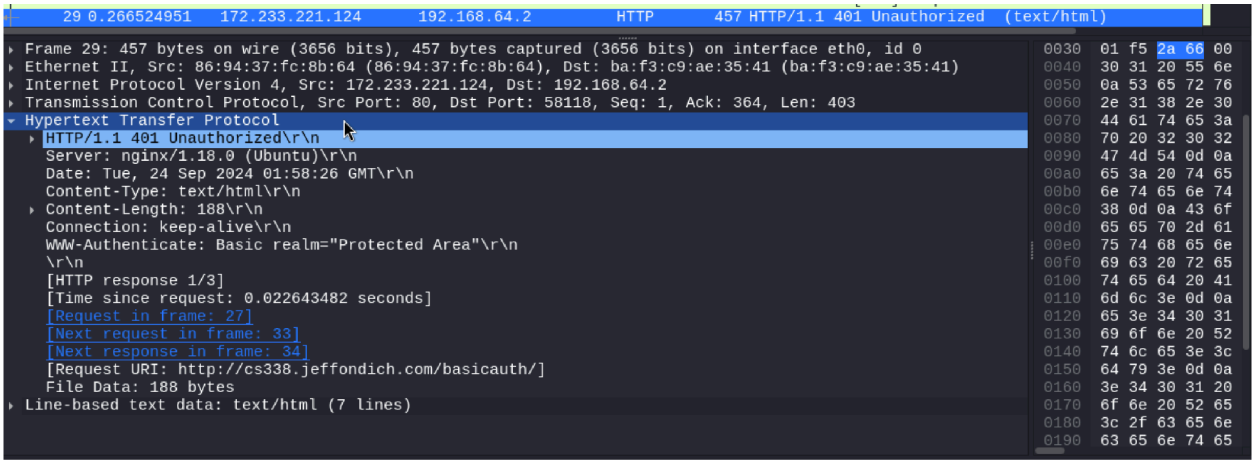
Sequence of events

TCP handshake

On line 19, a SYN packet is sent to initiate a connection from my kali linux (IP: 191.168.64.2) to the server hosting the http://cs338.jeffondich.com/basicauth/ (IP: 172.233.221.124) on port 80. The TCP handshake is completed (SYN, ACK, and final ACK), establishing a connection on lines 25-26.

HTTP GET request without authentication

On line 27, a *GET request is sent by my browser to access /basicauth/ without any authentication credentials* thus *the server responds with status code 401 Unauthorized* along with WWW-Aunthenticate header, indicating that credentials are required



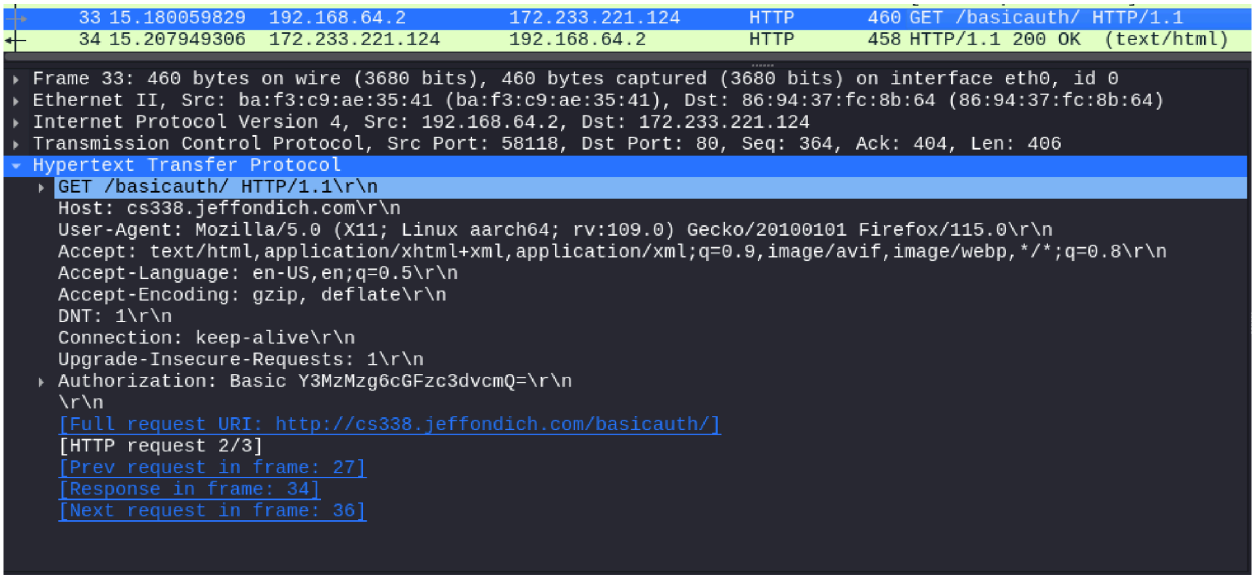
```
   29 0.266524951   172.233.221.124      192.168.64.2          HTTP       457 HTTP/1.1 401 Unauthorized  (text/html)

▸ Frame 29: 457 bytes on wire (3656 bits), 457 bytes captured (3656 bits) on interface eth0, id 0
▸ Ethernet II, Src: 86:94:37:fc:8b:64 (86:94:37:fc:8b:64), Dst: ba:f3:c9:ae:35:41 (ba:f3:c9:ae:35:41)
▸ Internet Protocol Version 4, Src: 172.233.221.124, Dst: 192.168.64.2
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 58118, Seq: 1, Ack: 364, Len: 403
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 401 Unauthorized\r\n
    Server: nginx/1.18.0 (Ubuntu)\r\n
    Date: Tue, 24 Sep 2024 01:58:26 GMT\r\n
    Content-Type: text/html\r\n
  ▸ Content-Length: 188\r\n
    Connection: keep-alive\r\n
    WWW-Authenticate: Basic realm="Protected Area"\r\n
    \r\n
    [HTTP response 1/3]
    [Time since request: 0.022643482 seconds]
    [Request in frame: 27]
    [Next request in frame: 33]
    [Next response in frame: 34]
    [Request URI: http://cs338.jeffondich.com/basicauth/]
    File Data: 188 bytes
▸ Line-based text data: text/html (7 lines)
```

```
0030   01 f5 2a 66 00
0040   30 31 20 55 6e
0050   0a 53 65 72 76
0060   2e 31 38 2e 30
0070   44 61 74 65 3a
0080   70 20 32 30 32
0090   47 4d 54 0d 0a
00a0   65 3a 20 74 65
00b0   6e 74 65 6e 74
00c0   38 0d 0a 43 6f
00d0   65 65 70 2d 61
00e0   75 74 68 65 6e
00f0   69 63 20 72 65
0100   74 65 64 20 41
0110   6d 6c 3e 0d 0a
0120   65 3e 34 30 31
0130   69 6f 6e 20 52
0140   74 6c 65 3e 3c
0150   64 79 3e 0d 0a
0160   3e 34 30 31 20
0170   6f 6e 20 52 65
0180   3c 2f 63 65 6e
0190   63 65 6e 74 65
```

HTTP GET request (with authorization)

After my browser receives the 401 response, I enter the credentials then the browser sends another *GET request to /basicauth/ including an Authorization header.*

Note: HTTP Basic Authentication encodes the username and password in base64 and sends it through the Authorization header. My credentials are a base64-encoded string of username:password thus Y3MzMzg6cGFzc3dvcmQ= decodes to css338:password in ASCII characters.

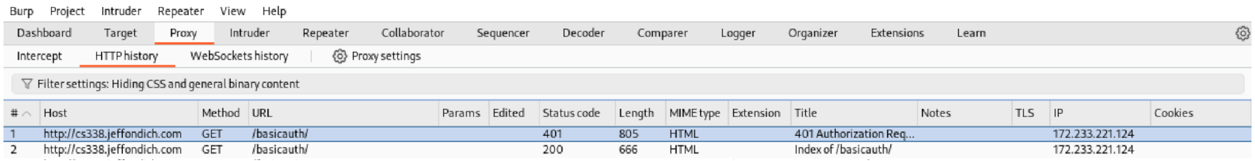


```
   33 15.180059829  192.168.64.2        172.233.221.124      HTTP       460 GET /basicauth/ HTTP/1.1
   34 15.207949306  172.233.221.124     192.168.64.2         HTTP       458 HTTP/1.1 200 OK  (text/html)

▸ Frame 33: 460 bytes on wire (3680 bits), 460 bytes captured (3680 bits) on interface eth0, id 0
▸ Ethernet II, Src: ba:f3:c9:ae:35:41 (ba:f3:c9:ae:35:41), Dst: 86:94:37:fc:8b:64 (86:94:37:fc:8b:64)
▸ Internet Protocol Version 4, Src: 192.168.64.2, Dst: 172.233.221.124
▸ Transmission Control Protocol, Src Port: 58118, Dst Port: 80, Seq: 364, Ack: 404, Len: 406
▾ Hypertext Transfer Protocol
  ▸ GET /basicauth/ HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  ▸ Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
    \r\n
    [Full request URI: http://cs338.jeffondich.com/basicauth/]
    [HTTP request 2/3]
    [Prev request in frame: 27]
    [Response in frame: 34]
    [Next request in frame: 36]
```

The *server then responds with HTTP/1.1 200 OK meaning that the authentication was successful* and the page can be accessed.

Note: since /basicauth/ only uses Basic Authentication, my credentials are only encoded and not encrypted thus can be easily decoded by anyone observing the communications (eg: Wireshark)

In summary:
TCP handshake (browser and server) → Initial GET request (browser to server) → 401 Unauthorized (server to browser) → User enters credentials → GET Request with Authorization header (browser to server) → 200 OK (server to browser).



Filter settings: Hiding CSS and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies
1	http://cs338.jeffondich.com	GET	/basicauth/			401	805	HTML		401 Authorization Req...			172.233.221.124	
2	http://cs338.jeffondich.com	GET	/basicauth/			200	666	HTML		Index of /basicauth/			172.233.221.124	

Burp suite screenshot